

Klassen en objecten hiervan vormen de basis van een object georiënteerde taal. In een puur object georiënteerde taal, zoals C#, is het zelfs onmogelijk buiten deze objecten te werken (cfr procedurele file in C++). Het grote voordeel van een OO taal is dat we het functionele probleem modulair kunnen aanpakken en ook makkelijker kan gemapt worden op de werkelijke wereld. In ons dagelijks (beroeps)leven zijn er ook verschillende objecten die met elkaar samenwerken.

Dit practicum verwacht dat je deze belangrijke basis onder de knie hebt en je dus klassen kan definiëren en instanties (objecten) hiervan kan aanmaken en gebruiken. De status van een object wordt bepaald door zijn data members, de functionaliteit dan weer door zijn functies (zie later). Verder hebben we gezien dat de toegang tot deze data members kan geregeld worden via properties (= data encapsulatie). Dit is een belangrijk gegeven in het bepalen van een correcte object status en moet dus steeds worden toegevoegd aan de klasse definitie.

### Opdracht 1

Maak een programma voor het beheren van een SAN server. Aan een SAN server kunnen 2 harde schijven worden toegevoegd + kan de RAID al dan niet worden ingesteld.

De volgende klassen zijn nodig:

#### HardDisk

- Eigenschappen
  - o Size (int): De size eigenschap geeft de grootte van de harde schijf aan. Een negatieve waarde wordt automatisch omgezet naar 0.
  - o Model (string): De model code van de harde schijf. Wanneer nog geen model is bepaald zal de volgende tekst worden getoond na het opvragen van de eigenschap:  
"\*\*\*\*\*"

#### SanServer

- Eigenschappen
  - o Harddisk1: data member van het type Harddisk
  - o Harddisk2: data member van het type Harddisk
  - o RaidSet: data member van het type bool (true indien RAID voorzien / false indien geen RAID voorzien)

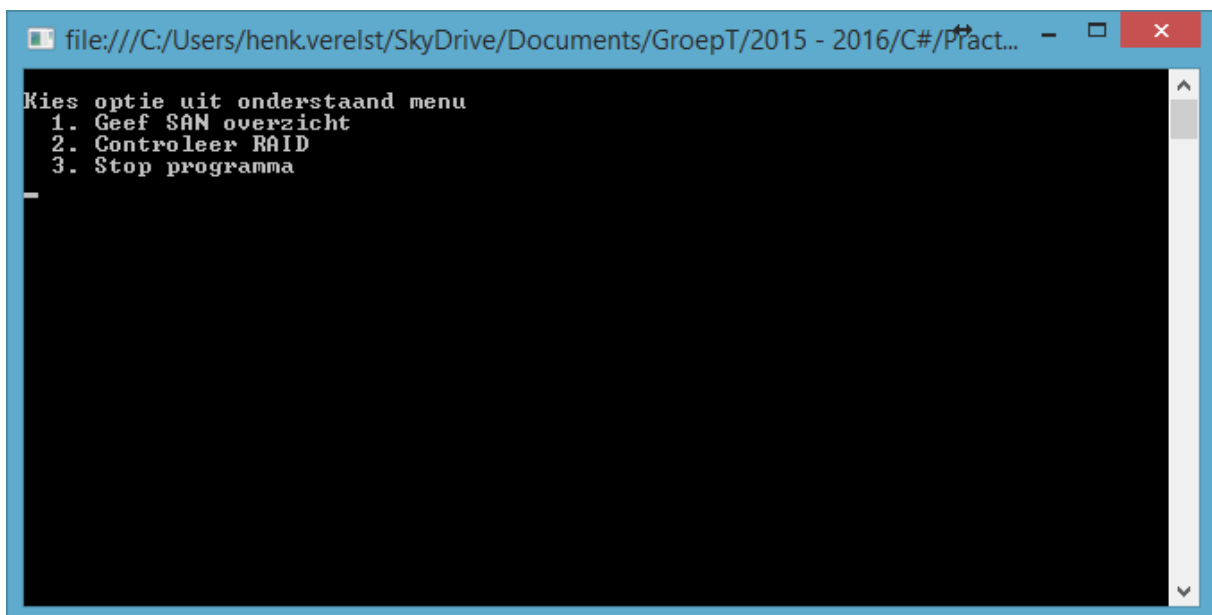
Hieronder een mogelijk flow van het programma:

1. Ingeven SAN gegevens



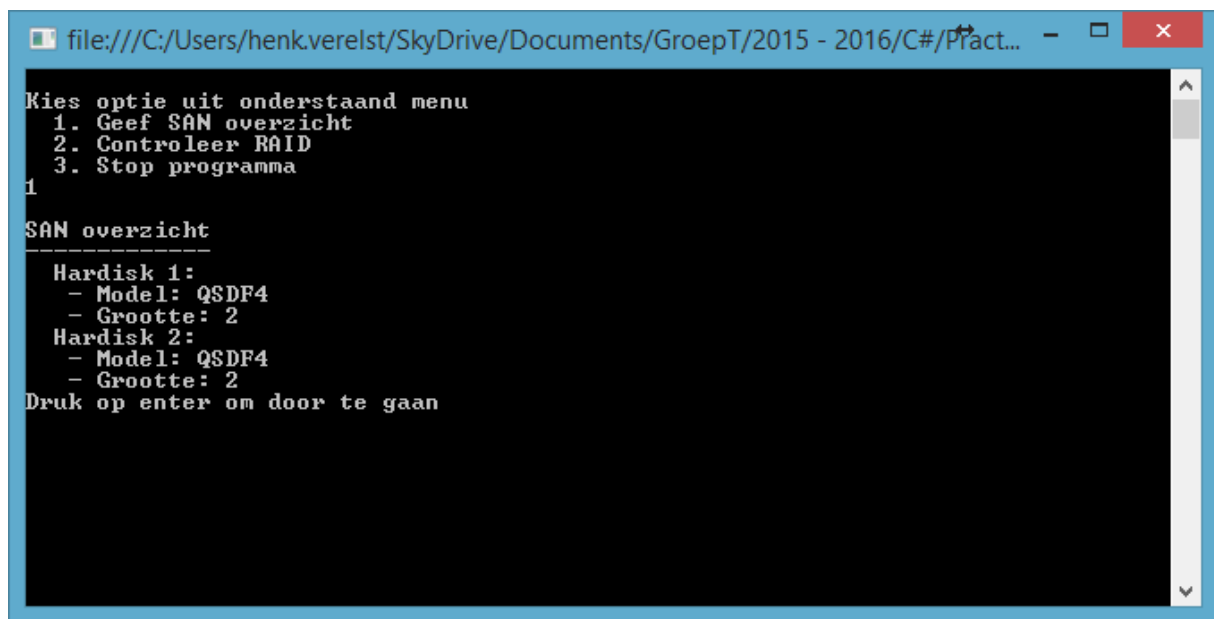
```
file:///C:/Users/henk.verelst/SkyDrive/Documents/GroepT/2015 - 2016/C#/Pract...
SAN BEHEER
=====
Geef de nodige SAN gegevens in
RAID ondersteuning <j/n>?: j
Gelieve de gegevens van hardisk 1 in te geven
Model: QSDP4
Grootte <Tb>: 2
Gelieve de gegevens van hardisk 2 in te geven
Model: QSDP4
Grootte <Tb>: 2
```

2. Controle SAN gegevens menu



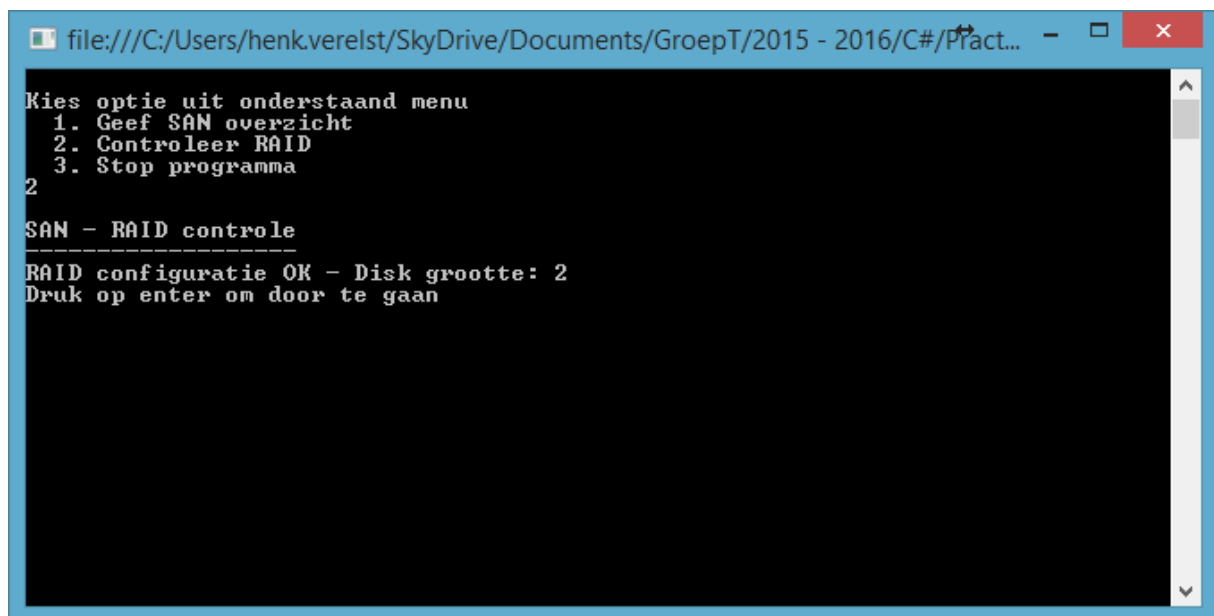
```
file:///C:/Users/henk.verelst/SkyDrive/Documents/GroepT/2015 - 2016/C#/Pract...
Kies optie uit onderstaand menu
1. Geef SAN overzicht
2. Controleer RAID
3. Stop programma
```

- a. Geef SAN overzicht



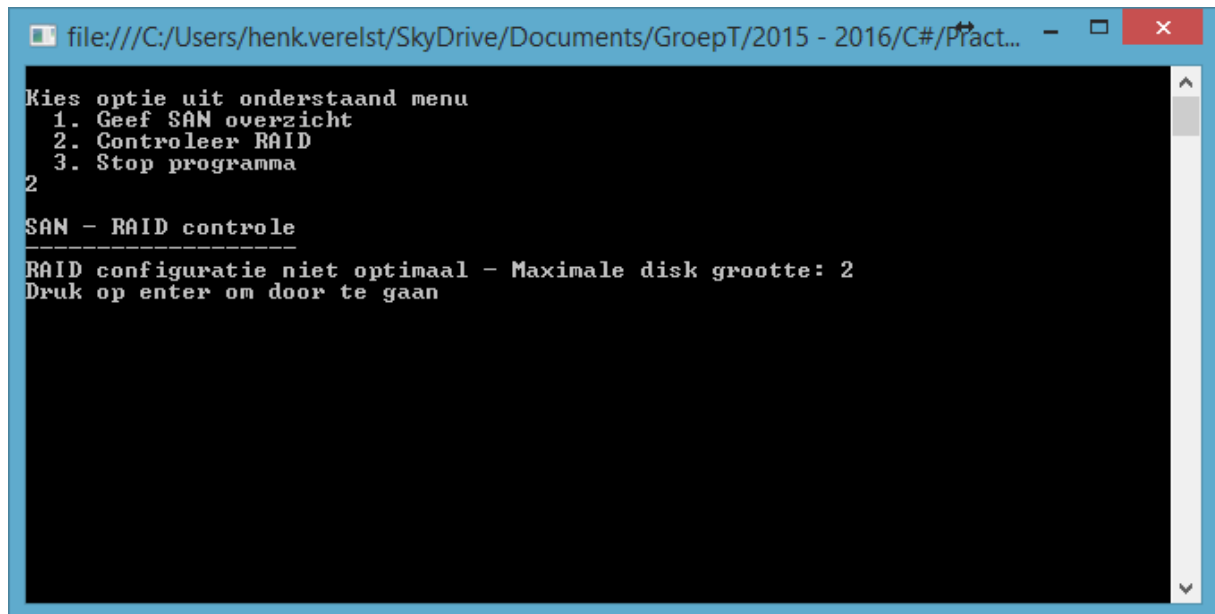
```
file:///C:/Users/henk.verelst/SkyDrive/Documents/GroepT/2015 - 2016/C#/Pract... - [X]
Kies optie uit onderstaand menu
1. Geef SAN overzicht
2. Controleer RAID
3. Stop programma
1
SAN overzicht
-----
Hardisk 1:
- Model: Q$DF4
- Grootte: 2
Hardisk 2:
- Model: Q$DF4
- Grootte: 2
Druk op enter om door te gaan
```

- b. Controleer RAID  
i. RAID ingesteld + disk size gelijk



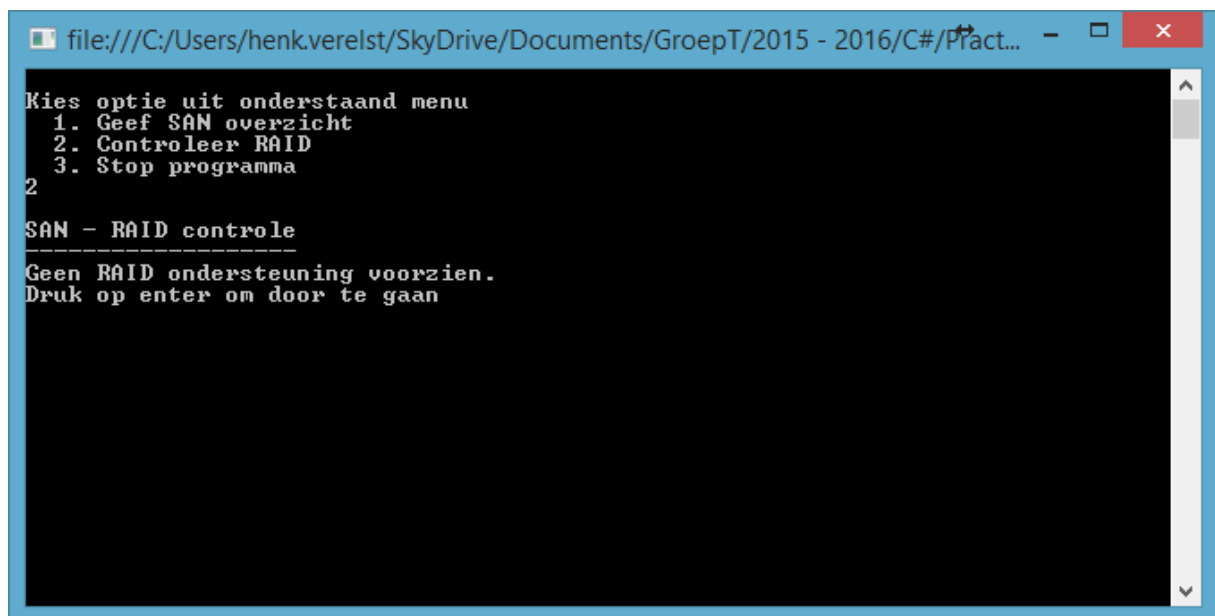
```
file:///C:/Users/henk.verelst/SkyDrive/Documents/GroepT/2015 - 2016/C#/Pract... - [X]
Kies optie uit onderstaand menu
1. Geef SAN overzicht
2. Controleer RAID
3. Stop programma
2
SAN - RAID controle
-----
RAID configuratie OK - Disk grootte: 2
Druk op enter om door te gaan
```

- ii. RAID ingesteld + disk size niet gelijk => disk met kleinste grootte bepaalt de SAN disk size (vb hd1 grootte = 2 / hd2 grootte = 4)



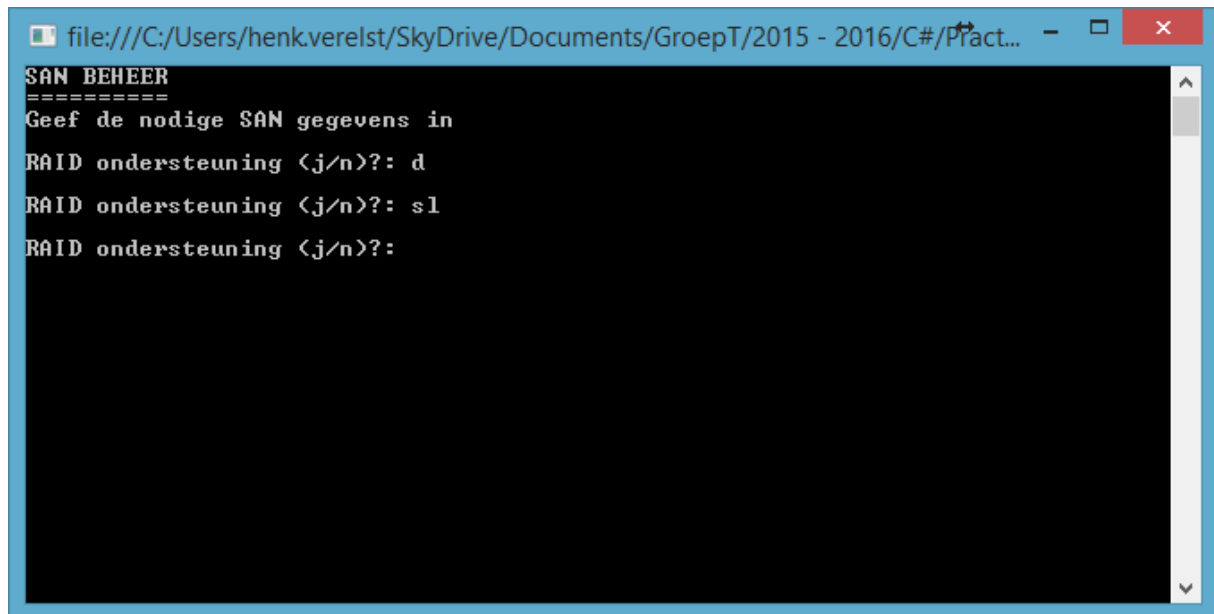
```
file:///C:/Users/henk.verelst/SkyDrive/Documents/GroepT/2015 - 2016/C#/Pract... - [X]
Kies optie uit onderstaand menu
1. Geef SAN overzicht
2. Controleer RAID
3. Stop programma
2
SAN - RAID controle
-----
RAID configuratie niet optimaal - Maximale disk grootte: 2
Druk op enter om door te gaan
```

- iii. RAID niet ingesteld



```
file:///C:/Users/henk.verelst/SkyDrive/Documents/GroepT/2015 - 2016/C#/Pract... - [X]
Kies optie uit onderstaand menu
1. Geef SAN overzicht
2. Controleer RAID
3. Stop programma
2
SAN - RAID controle
-----
Geen RAID ondersteuning voorzien.
Druk op enter om door te gaan
```

Indien de gebruiker geen geldige waarde ingeeft voor het bepalen van RAID ondersteuning wordt de vraag steeds opnieuw gesteld:



The screenshot shows a Windows file explorer window with the address bar displaying the file path: file:///C:/Users/henk.verelst/SkyDrive/Documents/GroepT/2015 - 2016/C#/Pract... The window contains a text file named 'SAN BEHEER'. The text inside the file is as follows:

```
SAN BEHEER
=====
Geef de nodige SAN gegevens in
RAID ondersteuning <j/n>?: d
RAID ondersteuning <j/n>?: s1
RAID ondersteuning <j/n>?:
```

Tip – Indien gewenst kan je het scherm na de menu keuze leeg maken met de volgende code regel:  
Console.Clear();